

Question Answering System Based on Artificial Intelligence for Restricted Domain

Meera Udani^{#1}, Avinash Shrivastava^{#2}, Vaibhav Shukla^{#3}, Archana Rao^{#4}

[#]Computer Engineering, Vidyalankar Institute of Technology, University of Mumbai, Maharashtra, India

Abstract— The project aims at an intelligent learning system that will take a text file as an input and extracts knowledge from the given text. Thus using this knowledge our system will try to answer questions fired to it by the user.

Question Answering(QA) systems are designed to find answers to open domain questions in a documents. The main goal of the Question Answering system is to encourage research into systems that return answers because many users prefer direct answers, and bring benefits of large-scale evaluation to QA task.

In its own form, the project could be used in online assessment of FAQs in different fields and also for interactive online lectures.

Keywords— Artificial Intelligence, Parts Of Speech (POS) Tagger, java

I. INTRODUCTION

Our system is designed to help people find specific answers to specific questions in restricted domain. It allows the user to pose questions in natural language and obtain relevant answers or assistance they require in order to solve certain tasks.

Natural Language Question Answering is recognized as a capability with great potential.

QA is different from the search engines in two aspects:

- (i) Instead of a string of keyword search terms, the query is a question in natural language, thus necessitating question processing
- (ii) Instead of a list of documents or URL's, answers at sentence level are expected to be returned in response to the query and thus a need for text processing supported by Natural Language Processing(NLP) and Information Extraction(IE).

One of the challenges of finding correct answers from the passages can be understood by the following example:

It is not difficult to match the question 'Who killed Abraham Lincoln?' with the text 'John Wilkes Booth killed Abraham Lincoln.' But it is more challenging to find the answer to the question in the text 'John Wilkes Booth is perhaps America's most infamous assassin. He

is best known for having fired a bullet that ended Abraham Lincoln's life.'

Difficulty of such problems can be minimized by the use of a dictionary to boost its performance.

The system contains three basic modules:

1. Question processing module
2. Document processing module
3. Answer Extraction and Formulation module

II. MOTIVATION AND BACKGROUND

Since our area was Artificial Intelligence a substantial amount of time was spent in gathering all the required information.

Question Answering is a computer science discipline within the fields of information retrieval and NLP which is concerned with building systems that automatically answer questions posed by humans in a natural language. A QA implementation, usually a computer program, may construct its answers by querying a structured database of knowledge or information, usually a knowledge base.

One of the earliest QA systems was ELIZA, developed in 1964. One successful ELIZA application was DOCTOR, a computer program that basically interacted with users through a text chat interface, answering questions and responding to the users dialog in a way that mimicked the client-centered psychotherapy between a client (the user) and their therapist (computer running the DOCTOR application).

Additional QA systems built on ELIZA include BASEBALL and LUNAR. BASEBALL answered questions about the US baseball league which spanned a period of one year. LUNAR answered questions about the geological analysis of rocks from the Apollo moon missions.

Both QA systems were very successful in their chosen domains

III. DESIGN AND WORKING

The architecture of the QA system as mentioned earlier, would consist of following three modules:

1. Question processing module.
2. Document processing module.
3. Answer extraction and formulation module.

The questions that the system receives can be divided into two major categories: FACTUAL & EXPERT.

Factual questions are those which contain words like *what, where, when, who*, etc.

Expert questions are those which contain words like *how, why* etc.

The level of intelligence required to solve expert questions is more than that of factual questions. We will be considering only factual questions.

The user is first asked to select the passage of his choice and then the type of question. The Question processing module will process the question and pass it to the Question Answering module which will make use of the various extractions received from the Document Processing phase, along with the Processed Documents containing the tagged format of the original input document. The tags can be generated using a POS tagger.

By applying required algorithms this module will pass it to the Formulation module for getting the desired answer.

The 3 modules have been implemented in java using the following 3 functions.

The Document Processing Module was implemented using the following function:

Description:

- It takes in the choice of the user for a particular passage from the displayed list and using file reader reads it and using StringTokenizer tokenizes it
- Then using POS Tagger tags the tokens
- With the help of tags finds the verbs in the passage
- Using the list of irregular verbs and the logic for regular verbs a data structure (array) is created which contains the verbs along with their tenses and *ing* form.
- Along with the verb array we also find the actors from the passage

The Question Processing Module was implemented using the following function:

Description:

- It takes in a question from the user
- Using StringTokenizer tokenizes it and stores it in another data structure (array)
- Returns it for further use in the program

The Question-Answering Module was implemented using the following function:

Description:

- First finds the verb in the question
- Matches the verb just found with the tokens created in the document processing stage
- According to the selected case for type of factual question (*what, when*, etc) it further tries to extract and formulate the answer

The entire functioning can be explained by the figure shown below

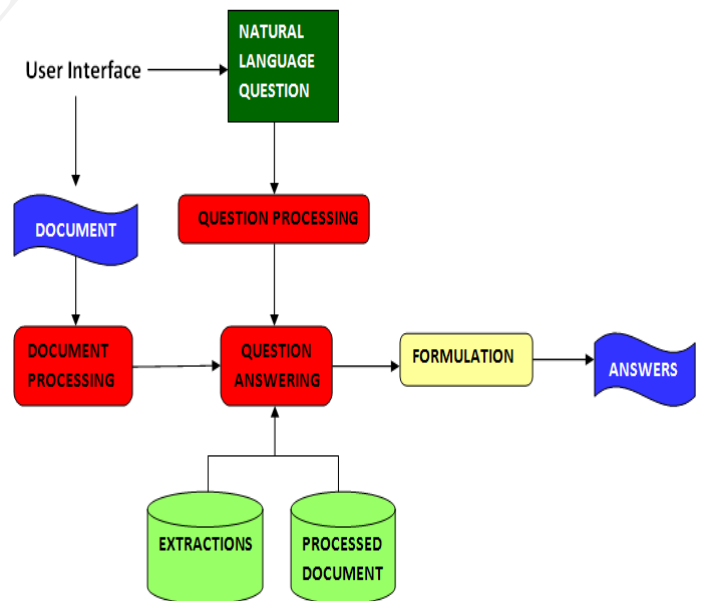


Figure 1: System flow diagram

IV. APPLICATIONS AND FUTURE SCOPE

We have described a system that handles a question in natural language and tries to provide its answer in a single statement.

With certain improvements; the proposed system can be used for the following applications:

- Can be used in situations where a quick review of an entire text takes time.
- Adding speech recognition abilities to the current system will enable people with reading disabilities to take advantage of this system.
- This system can be used to make online lectures more old-school type by allowing lectures to proceed only when questions related to the previous lecture are answered correctly.
- The QA paradigm extends beyond AI systems to query processing in database systems and many analytical tasks that involve gathering, correlating and analysing information; can naturally be formulated as QA problems.
- The existing system can be integrated with a search engine to enhance the performance.

V. CONCLUSION

We have described a system that tries to process a question asked in natural language related to a particular passage and answers it in a more human-like manner as possible using Artificial Intelligence rather than the traditional NLP. Overall success is limited, because firstly, answering is restricted to a precise domain and secondly, user has to follow a particular format while entering a passage and asking related questions.

ACKNOWLEDGMENT

We sincerely thank Prof. Avinash Shrivastava, Computer department, Vidyalankar Institute of Technology, Mumbai for his valuable guidance and motivation for this work.

REFERENCES

- [1] Abhijit Kumar and Dr. Lavit Rawtani, "Question Answering System using Artificial Intelligence and Fuzzy Systems"
- [2] Maria Vargas-Vera and Enrico Motta, "AQUA: A Question Answering System for Heterogeneous Sources".
- [3] Artificial Intelligence by Stuart Russel & Peter Norvig.
- [4] www.ieee.org -site for IEEE papers.